Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

A Segmentation based Adaptive Approach for Cursive Handwritten Text Recognition

Brijesh Verma and Hong Lee

Abstract—**The paper presents a segmentation based adaptive approach for the learning and recognition of single person's handwritten text. The approach is incorporated into an automated intelligent system for scanning of handwritten text on a paper and converting it into a text file. It scans an A4 size handwritten page and segments it into lines, words and characters. The segmented characters are passed to a neural classifier for the recognition. The final word is passed through a lexicon based matching process to improve the accuracy of the recognized text. Two neural networks are investigated for the learning of segmented characters quickly and accurately. The experimental results show that the proposed approach can produce high text recognition accuracy with a small number of training samples.**

I. INTRODUCTION

The recognition of handwritten text is a difficult and challenging task in many real world applications [1-3] including postal address recognition, bank check recognition, document authentication, form processing and restoring old archives.

In the last few decades, many papers and book chapters describing new techniques for the classification of handwritten numerals [4-8], characters [9-13] and words [14-31] have been published. Despite the promising results for isolated numerals and characters using conventional and intelligent techniques, the successful conversion of scanned handwritten page into text is still not possible. The existing commercial systems are unable to recognize handwritten text with satisfactory results and they have very limited capability to recognize printed characters and text. There is no commercial system available which can scan a handwritten page and convert it into text with an acceptable accuracy. The researchers and developers in the area of handwriting recognition systems have mainly focused on developing a general system which can recognize any person's handwriting. However, acceptable results have not been achieved yet. There are many reasons for obtaining low text recognition rates such as (i) attempt to develop a general system for any type of handwritten text (ii) cursive and touching nature of handwritten text (ii) inaccurate segmentation of handwritten text (iii) ambiguous and incomplete segmented characters and (iv) inaccurate feature extraction, just to mention a few. All reasons for unsatisfactory performance for text recognition point us to one conclusion which is low accuracy on characters segmented/extracted from handwritten text. The low accuracy on segmented characters is obtained because of low generalization abilities of classifiers which is caused by number of samples used for training and the type of learning algorithm used. The training of neural learning algorithms is usually conducted using many thousands of handwritten samples. It is common perception in industry that if we use intelligent techniques such as neural classifiers, we need a very large number of training samples to produce a good generalization or high accuracy on test samples. However, this perception has not been investigated properly for the recognition of single person's handwriting. Therefore, in this paper we focus on investigation of two major issues as follows (i) the number of samples required for a classifier to achieve a high text recognition accuracy (ii) training and testing of the system for single person's handwritten text. The main aim of this research is to develop a system which can be installed on a single-user machine, learn and recognize single person's handwritten text with an acceptable accuracy.

The remainder of this paper is broken down into four sections. Section II presents the proposed handwritten text recognition approach, Section III provides experimental results, a discussion of the results takes place in Section IV, and finally Section V presents conclusions.

II. PROPOSED HANDWRITTEN TEXT RECOGNITION APPROACH

An overview of the proposed handwritten text recognition approach is shown below in Figure 1 and it is described in detail in sub-sections A-G.

A. Scanning

An A4 size page with handwritten text is scanned in jpeg format and converted into a pbm (pixel bit map) format for further processing. The pbm format contains image width, image height and raw pixels in black/white (0/1) which makes processing such as shifting, resizing and segmentation of handwritten text much easier and faster.

B. Verma is with Central Queensland University, Australia (phone: 61749309058; fax: 61749309027; e-mail: b.verma@ cqu.edu.au).

H. Lee is with Central Queensland University, Australia (e-mail: h.lee@.cqu.edu.au).

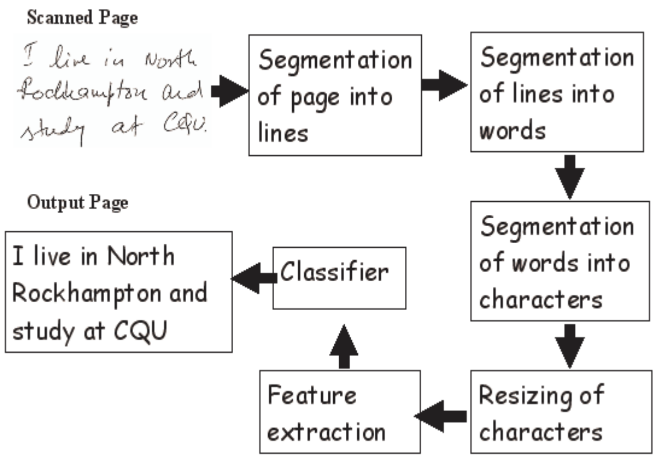

Fig. 1. Overview of the proposed approach

B. Segmentation

Segmentation of the text into lines is conducted based on horizontal histograms. The lines are segmented with lowest density (highest number of 0s between two lines). Segmentation of lines into words is conducted based on low vertical pixel density. It is assumed that the space between the words is greater than the space between the characters. The segmentation of words into characters is one of the most difficult processes [14-22] in text recognition. The word segmentation in this research is based on over-segmentation and selection of correct points [14]. The points from the areas as holes, cavities, etc to avoid cutting characters such as a, o, v, w into half, are removed. The word segmentation algorithm is briefly described below and the examples of segmented words are shown in Figure 2.

Algorithm 1: Segmentation

Step 1: The baselines such as upper, lower and middle baselines are calculated.
Step 2: The word is over-segmented.
Step 3: The obtained segmentation points are passed through the various rules.
Step 4: The incorrect segmentation points are removed.
Step 5: The correct segmentation points are used for final segmentation of the words.

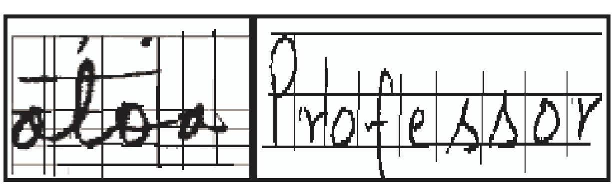

Fig. 2. Segmentation of words

C. Shifting and resizing of characters

The position and size of a character is very important for feature extraction and classification processes. We propose and use the following simple algorithms for shifting the character on top-left corner and resizing the character image.

Algorithm 2: Removing 0s of upper and left side of the character

Step 1: Start at top-left corner.
Step 2: Calculate horizontal and vertical densities.
Step 3: Start at row 1 and remove all rows from top until horizontal density is 0.
Step 4: Start at column 1 and remove all columns from left until vertical density is 0.
Step 5: Store new image width and height.

Algorithm 3: Resizing of character image

Step 1: Calculate ratio of real image width/target image width.
Step 2: If ratio is positive then shrink the image by removing the columns otherwise increase the image by filling the columns.
Step 3: Repeat steps 1-2 for rows using the height of the image.

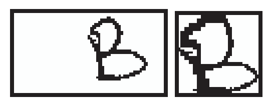

Fig. 3. Shifted (top-left corner) and resized character B

D. Feature extraction

Feature extraction is an important component of the segmentation-based approaches for handwritten text recognition. However, the extraction of appropriate features for segmented characters and features during the segmentation is a very difficult task. In the past few decades, many techniques such as density feature, direction feature, contour feature, transition feature, etc. have been investigated [14, 24, 26-28]. We used transition information [28] to extract appropriate feature values from segmented and resized characters. The technique is based on the calculation and location of transitions from background to foreground pixels in the vertical and horizontal directions.

E. Preparation of training and testing data

A program was written in C++ which can read characters from pbm file and automatically store features (100 float values) extracted for characters and corresponding output in forms of 52 values (1 for each character). For example, “a” is stored as follows:

Input (100 feature values)
0.51 0.13 0.00 0.00 0.00 0.69 0.22 0.00 0.00 0.00 0.82 0.27 0.04 0.00 0.00 0.72 0.21 0.00 0.00 0.00 0.77 0.32 0.10 0.00 0.00 0.78 0.27 0.00 0.00 0.00 0.88 0.42 0.00 0.00 0.00 0.82 0.39 0.09 0.00 0.00 0.77 0.17 0.00 0.00 0.00 0.77 0.37 0.13 0.03 0.00 0.94 0.93 0.86 0.89 0.93 0.17 0.59 0.21 0.24 0.25 0.00 0.11 0.00 0.07 0.10 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.38 0.68 0.72 0.54 0.48 0.02 0.12 0.13 0.26

0.16 0.00 0.03 0.00 0.01 0.03 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

Output (52 values for upper & lower case characters)
0.90 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10 0.10

F. Neural classifier

We incorporated two neural classifiers in the proposed approach, first one is based on modified gram-schmidt (MGS) algorithm and second one is based on standard back propagation (BP) algorithm. For experimentation purposes, the architectures were modified varying the number of inputs, outputs, hidden units, etc. The modified gram-schmidt method is very fast and automatically finds the best number of hidden units. It is fast because it directly calculates the weights in contrast to iterative process in BP. The weights of hidden units are randomly set. After setting the weights the values of hidden units (inputs to output layer) are calculated for every input and stored in a matrix form (eg. X). The matrix X and the target output are used to calculate the weights of the output layer. The MGS is used to calculate the weights.

G. Recognition of words using lexicon

A lexicon matching process was implemented to match recognized words with words stored in a lexicon. An algorithm based on character by character matching and assigning a confidence is proposed and used in this research.

III. Experimental results

The proposed approach presented in Figure 1 has been implemented and the experiments by incorporating Modified Gram Schmidt (MGS) and Back Propagation (BP) algorithms for weight adjustment of our learning technique were conducted. In our previous handwriting recognition research we have used CEDAR benchmark database with large number of characters, however in this research we did not use CEDAR or any other benchmark database because the aim is to train and test the proposed approach with only one individual's handwritten text. The idea is to install the system on individual's platform and train it with his/her handwriting so that it can be used by that particular individual and produce high text recognition accuracy. The aim of the experiments is to test the performance of our proposed approach and find out how many sets of lower and upper cases characters are needed from one individual for training to achieve a satisfactory performance by the proposed approach.

Many sets of experiments using training and testing samples of 52, 104, 156, 204, 256 and 312 characters were conducted. Some training and testing samples are presented below in Figure 3. The characters as shown below in Figure 3 were used to train the proposed approach and handwritten pages such as one shown below were used for testing. Table I and Table II present the results for characters and text recognition.

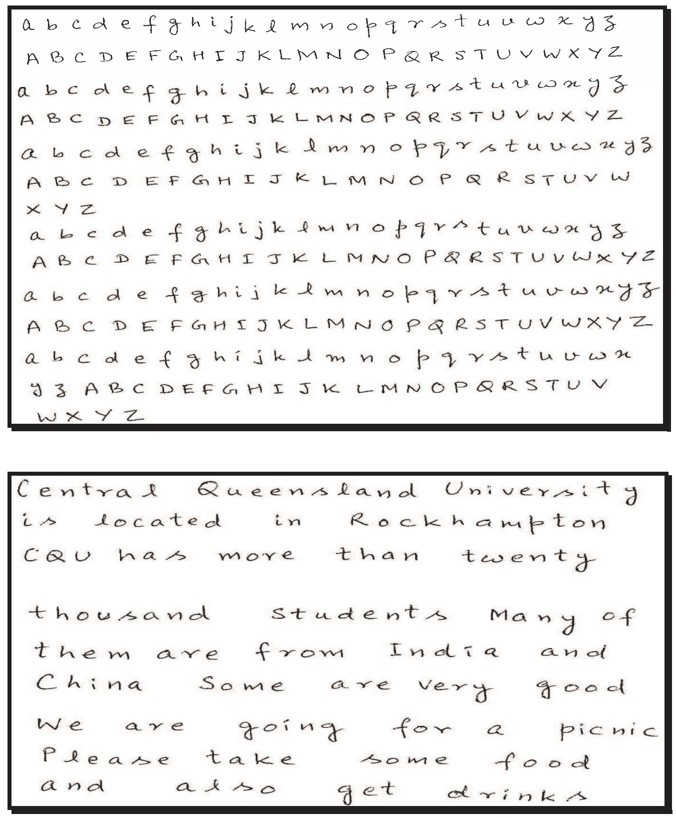

Fig. 3. Samples of training and test examples

TABLE I
CLASSIFICATION ACCURACY FOR SEGMENTED CHARACTERS

Approach	Sample Size (Number of Characters)	Character Classification Accuracy [%]		
		lower	upper	both
MGS	52	42.30	26.92	36.53
MGS	156	69.23	84.61	84.61
MGS	208	80.77	88.46	88.46
MGS	312	92.31	80.77	92.31
BP	52	57.69	69.23	71.15
BP	156	65.38	84.61	80.76
BP	208	84.61	84.61	86.53
BP	312	96.15	88.46	96.15

TABLE II
CLASSIFICATION ACCURACY FOR TEXT RECOGNITION

Approach	Sample Size (Number of Characters)	Text Classification Accuracy [%]
MGS	52	21.42
MGS	156	71.42
MGS	208	85.71
MGS	312	100.00
BP	52	46.15
BP	156	71.42
BP	208	78.57
BP	312	100.00

IV. Analysis and Discussion

The results for handwritten characters as well as text are shown in Tables I and II. The character classification accuracy over 96% has been achieved. As expected, the classification accuracy improved with the increase of training samples. The text recognition accuracy for various test pages was also very high. The text recognition rate was also improved with the increase of training samples which is not a surprise but it was very interesting to see that with only 312 characters, acceptable accuracy was achieved. In fact all words were recognized which gave 100% accuracy. During the analysis, we noticed that many words were recognized with small number of characters in it which means that the 100% character classification accuracy is not required to obtain 100% text accuracy. It was also noticed that some simple characters from words were not recognized but when we cut those characters from words/text and passed to our system then they were perfectly recognized by the system. It appears that during the segmentation of words into characters some characters are left with a large number of zeros and shifted towards one end which might have caused the problem, the problem is being currently investigated.

There were some common character misclassification errors, for example the character c (lower case) was recognized as C (upper case) and O (upper case) recognized as o (lower case). Some examples of common errors are shown below in Table III.

TABLE III
COMMON MISCLASSIFICATION ERRORS

Original	c	g	i	o	t	z	C	M	O	Z
Recognized	C	z	I	O	y	y	C	H	o	e

During the text recognition's final stage, sometimes the lexicon matching process in our approach has retrieved the correct word, even if the number of characters recognized was not very high. Some examples of the recognized word using 52 and 208 samples are shown below in Table IV.

TABLE IV
EXAMPLES OF RETRIEVED WORDS

Word#	52 samples	208 samples	Correct word
1	BfhjZOk	BrjjeJh	Brijesh
2	VlTmI	VBrmX	Verma
3	CDU	CQU	CQU

We have analyzed the character and text accuracies in relation to number of characters. A graph with the character and text accuracies by varying the number of characters is presented in Figure 4.

The experimental results from this research have been compared with the results from existing text recognition software. Most of the existing software such as FormStorm, ABBYY, OmniPage and ritePen recognize only printed characters or online handwriting. Some does not allow testing on own datasets. Therefore a comparison with the existing systems is not an easy task. We tried to compare the proposed approach with 3 existing systems such as "Simple software", "HP director" and "Readiris Pro 11". Simple software required larger training data set but gave much worst results than the proposed approach. The test results from HP scanner using our test data were very poor. There is no option for training so it was only used to test on trained parameters which might not be suitable for handwriting used in test data and this might be the reason for poor accuracy. Readiris Pro 11 is quite accurate for machine printed letters achieving correct conversion rate of around 95%. However, hand-printing recognition was a different story, which gives around 70% ~ 76% conversion rates. The accuracy on our test data vary between 68.5% and 73.5%. Conclusively, it could be said that the average character recognition rates would be around 71%. It was not possible to get word/text recognition accuracy from Readiris Pro 11. The comparative results are shown in Table V.

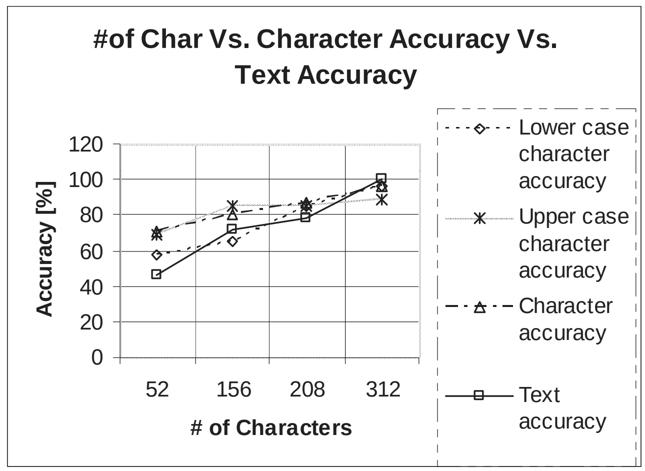

Fig. 4. Character vs. text accuracy

TABLE V
COMPARISON OF TEXT ACCURACY RESULTS

Approach	Training	Test [%] handwritten pages
Proposed Approach	312 chars	100
Simple Software	300 words	86.76

V. Conclusion

We have presented a novel approach based on two neural networks for the recognition of handwritten text. The approach produces over 96% character classification accuracy and 100% text recognition accuracy in just a few minutes of training. The investigations showed that the proposed approach requires only 312 characters to achieve satisfactory text recognition accuracy. The training with such a small number of character samples was very quick so the approach is practical for use in real world applications.

We must mention here that the handwritten pages used for testing contained easy handwritten text in terms of spaces between lines, words and characters. More experiments are required to test the approach on difficult touching handwritten text.

REFERENCES

[1] M. N. Kapp, C. de Almendra Freitas and R. Sabourin, Methodology for the Design of NN-based Month-Word Recognizers Written on Brazilian Bank Checks, Image Vision Computing, vol. 25, no. 1, 2007, pp. 40-49.

[2] Q. Xu, L. Lam and C. Y. Suen, "Automatic Segmentation and Recognition System for Handwritten Dates on Canadian Bank Cheques", Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR), 2003.

[3] Y. Wen, Y. Lu, and P. Shi, Handwritten Bangla Numeral Recognition System and Its Application to Postal Automation, Pattern Recognition, vol. 40, no. 1, pp. 99-107, 2007.

[4] D. Gorgevik and D. Cakmakov, An Efficient Three-Stage Classifier for Handwritten Digit Recognition, ICPR, vol. 4, 2004, pp. 507-510.

[5] S-W. Lee, "Off-Line Recognition of Totally Unconstrained Handwritten Numerals Using Multilayer Cluster Neural Network", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 18, 1996, pp. 648-652.

[6] S-B. Cho, "Neural-Network Classifiers for Recognizing Totally Unconstrained Handwritten Numerals", IEEE Trans. on Neural Networks, vol. 8, 1997, pp. 43-53.

[7] S-W. Lee, "Multilayer Cluster Neural Network for Totally Unconstrained Handwritten Numeral Recognition", Neural Networks, vol. 8, 1995, pp. 783-792.

[8] C. Y. Suen and J. Tan, "Analysis of Errors of Handwritten Digits Made by a Multitude of Classifiers", Pattern Recognition Letters, Vol. 26, No. 3, 2005, pp. 369-379.

[9] H. I. Avi-Itzhak, T. A. Diep and H. Garland, "High Accuracy Optical Character Recognition using Neural Networks with Centroid Dithering", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 17, 1995, pp. 218-224.

[10] N. Arica and F.T. Yarman-Vural, "Optical Character Recognition for Cursive Handwriting", IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, 2002, pp. 801-813.

[11] F. Camastra and A. Vinciarelli, "Combining Neural GAs and Learning Vector Quantization for Cursive Character Recognition", Neurocomputing, vol.51, 2003, pp. 147-159.

[12] M. Hanmandlu, K. R. M. Murali, S. Chakraborty, S. Goyal and D. R. Choudhury, "Unconstrained Handwritten Character Recognition based on Fuzzy Logic", Pattern Recognition, vol.36, 2003, pp. 603 – 623.

[13] A. de S. Britto Jr, R. Sabourin, F. Bortolozzi and C. Y. Suen, "Foreground and Background Information in an HMM-based Method for Recognition of Isolated Characters and Numeral Strings", Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition , IWFHR, 2004.

[14] Verma, B. "A Contour Code Feature Based Segmentation For Handwriting Recognition", 7th IAPR International Conference on Document Analysis and Recognition, ICDAR'03, 2003, pp. 1203-07.

[15] N.W. Strathy, C.Y. Suen and A. Krzyzak, "Segmentation of Handwritten Digits using Contour Features", ICDAR '93, 1993, pp. 577-580.

[16] R. G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, 1996, pp. 690-706.

[17] C. E. Dunn and P. S. P. Wang, "Character Segmentation Techniques for Handwritten Text - A Survey", Proceedings of the 11th International Conference on Pattern Recognition, The Hague, The Netherlands, 1992, pp. 577-580.

[18] Y. Lu and M. Shridhar, Character Segmentation in Handwritten Words - An Overview, *Pattern Recognition*, vol. 29, 1996, pp. 77-96.

[19] H. Fujisawa, Y. Nakano and K. Kurino, Segmentation Methods for Character Recognition: from Segmentation to Document Structure Analysis, Proceedings of the IEEE, vol. 80, 1992, pp. 1079-1092.

[20] G. Dimauro, S. Impedovo, G. Pirlo and A. Salzo, An Advanced Segmentation Technique for Cursive Word Recognition, Advances in Handwriting Recognition, S. W. Lee (ed.), World Scientific Publishing, 1998, pp. 255-264.

[21] X. Xiao and G. Leedham, Knowledge-based Cursive Script Segmentation, Pattern Recognition Letters, vol. 21, 2000, pp. 945-954.

[22] Y. Sun , T. S. Butler , A. Shafarenko , R. Adams , M. Loomes and N. Davey, Word Segmentation of Handwritten Text Using Supervised Classification Techniques, Applied Soft Computing, vol. 7, no. 1, 2007, pp.71-88.

[23] A. Bensefia , T. Paquet, and L. Heutte, A Writer Identification and Verification System, Pattern Recognition Letters, vol. 26 no. 13, 2005, pp. 2080-2092.

[24] R. M. Bozinovic and S. N. Srihari, "Off-Line Cursive Script Word Recognition", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 11, 1989, pp. 68-83.

[25] J-H. Chiang, "A Hybrid Neural Model in Handwritten Word Recognition", Neural Networks, vol. 11, 1998, pp. 337-346.

[26] B. Eastwood, A. Jennings and A. Harvey, "A Feature Based Neural Network Segmenter for Handwritten Words", International Conference on Computational Intelligence and Multimedia Applications, Australia, 1997, pp. 286-290.

[27] S. Gunter and H. Bunke, "Feature Selection Algorithms for the Generation of Multiple Classier Systems and Their Application to Handwritten Word Recognition", Pattern Recognition Letters, vol. 25, 2004, pp. 1323-1336.

[28] P. D. Gader, M. Mohamed and J-H. Chiang, "Handwritten Word Recognition with Character and Inter-Character Neural Networks", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 27, 1997, pp. 158-164.

[29] C. Viard-Gaudin, P-M. Lallican and S. Knerr, "Recognition-directed Recovering of Temporal Information from Handwriting Images", Pattern Recognition Letters, vol. 26, no. 16, 2005, pp. 2537-2548.

[30] M.-P. Schambach, "Fast Script Word Recognition with Very Large Vocabulary", 8th International Conference on Document Analysis and Recognition, 2005, pp. 9-13.

[31] S. Chevalier, E. Geoffrois, F. Preteux, and M. Lemaltre, "A Generic 2D Approach of Handwriting Recognition", 8th International Conference on Document Analysis and Recognition, 2005, pp. 489-493.